Operations Research Techniques and Algorithms (620-361)

Dr Yao-ban Chan

y.chan@ms.unimelb.edu.au
Telephone: 8344 9073
Office: Room 198, Richard Berry Building

Christina Burt
c.burt@ms.unimelb.edu.au
Telephone: 8344 1797
Office: 139 Barry St

Friday 11th April 2008

Today's Lecture

Unimodal n-D unconstrained optimisation
- Non-derivative methods

Constrained optimisation
- Optimality conditions

Non-derivative methods The algorithms discussed so far for finding unconstrained minima of functions of n variables require Hessian and/or gradient calculations:

- The Steepest Descent Method requires the calculation of $\nabla f(x)$
- Newton's Method requires the calculation of $\nabla f(x)$ and $\nabla^2 f(x)$
- Quasi-Newton Methods require the calculation of $\nabla f(x)$

Recall the line-search methods for finding minima of functions of one variable. The Fibonacci Search and Golden Section search algorithms did not require any derivative calculations. Are there non-derivative methods for finding minima of functions of n variables ? *Yes*:

Non-derivative methods

- The Nelder-Mead "Simplex Method" (not to be confused with Simplex Method for solving LPs)
- The coordinate descent method
- Methods which approximate derivatives using finite difference formulae

Stochastic optimisation If f is unimodal, then descent methods are generally able to find the (unique) minimum of f.

If f has multiple minima, for example, some local minima and an absolute minimum, then descent methods will find *one* of these for each application of the method (we may get different minima with different initial iterates x^0).

Usually, we are interested in attaining global minima, and we wish to avoid getting "stuck" in local minima. One approach is to try a range of different initial iterates x^0.

Alternative: stochastic search methods, which have some positive probability of searching in a non-descent direction. A classic example is the method of Simulated Annealing.

Today's Lecture

Unimodal n-D unconstrained optimisation
- Non-derivative methods

Constrained optimisation
- Optimality conditions

Constrained Optimization

We assume throughout this chapter that $f : \Re^n \to \Re$ is a C^1 objective function.

For $p, q \in \{0, 1, 2, \ldots\}$, let

$$g_1, \ldots, g_p,\ h_1, \ldots, h_q$$

be C^1 functions from $\Re^n$ to $\Re$ and

$$g(x) = (g_i(x))_{i=1}^p \text{ and } h(x) = (h_j(x))_{j=1}^q,$$

so $g : \Re^n \to \Re^p$ and $h : \Re^n \to \Re^q$ are C^1 vector-valued functions.

The problem of interest is the *nonlinear program*,

$$\begin{array}{ll} \min\limits_{x} & f(x) \\ \text{subject to} & g(x) \leq 0, \quad h(x) = 0, \end{array} \tag{NLP}$$

where the vector inequality $g(x) \leq 0 \in \Re^p$ means $g_i(x) \leq 0$ for each $i = 1, \ldots, p$, and the vector equality $h(x) = 0$ means $g_j(x) = 0$ for each $j = 1, \ldots, q$.

If $p = 0$, (NLP) is an equality-constrained problem.
If $p = q = 0$, (NLP) is unconstrained and the first-order necessary condition for x^* to minimize f would be that $\nabla f(x^*) = 0$.
We will generalize this stationarity condition to obtain first-order necessary conditions for (NLP). First we review the easier case of nonlinear programs with only equality constraints.

Optimality conditions for equality-constrained optimization

The equality-constrained NLP is

$$\begin{array}{ll} \min_x & f(x) \\ \text{subject to} & h(x) = 0. \end{array} \tag{1}$$

This is just (NLP) with no inequality constraints, $p = 0$. Lagrange gave the first-order necessary conditions for this problem.

The *Lagrangian* function for (1) is

$$L(x,\eta) \;:=\; f(x) + \sum_{j=1}^{q} \eta_j h_j(x) \;=\; f(x) + \langle \eta, h(x) \rangle,$$

where $\eta \in \Re^q$. The vector η is called the *Lagrange multiplier* corresponding to $h(x)$, in fact each component η_j of η is a multiplier corresponding to each component $h_j(x)$ of $h(x)$.

Suppose that either

- h is an affine (linear plus constant) function, or
- the set of gradients $\{\nabla h_j(x) : \text{all } j\}$ is linearly independent.

Each of these conditions is called a *constraint qualification*, and ensures that h is not "badly behaved".

The first-order necessary conditions for x^* to be a local minimum of (1) are given in the following theorem.

Theorem 4: *Let f and h be C^1 functions and assume that either of the above constraintqualifications hold at x^*. If x^* is a local minimimum of (1) then there exists $\eta^* \in \Re^q$ such that*

$$\begin{aligned} 0 &= \nabla_x L(x^*, \eta^*) \\ 0 &= h(x^*). \end{aligned}$$

The vector η^* is called an *optimal* (Lagrange) multiplier for x^*.

The first equation reflects the fact that the Lagrangian must be stationary with respect to x, the second equation requires x^* to be feasible.

Observe that the gradient of L is

$$\nabla_x L(x^*, \eta^*) \; = \; \nabla f(x^*) + \sum_j \eta_j^* \nabla h_j(x^*) \; = \; \nabla f(x^*) + \nabla h(x^*)\eta^*,$$

where the matrix $\nabla h(x^*)$, known as the *Jacobian*, is an $n \times q$ matrix.

The fact that local optima are stationary points of the Lagrangian function with respect to x in the case of a single equality constraint can be seen by looking at the level curves of f and the feasible space.

At the optimum, the level curves must be parallel to the feasible region (if not, then it is possible to travel inside the feasible region in a way which decreases f). Therefore their normals must be parallel:

$$\nabla f(x^*) \propto \nabla h(x^*)$$

$$\nabla f(x^*) = -\eta^* \nabla h(x^*)$$

$$\nabla f(x^*) + \eta^* \nabla h(x^*) = 0$$

(remembering that the gradient is taken with respect to the x's only).
This is the Lagrange condition.

Consider the equality-constrained NLP

$$\begin{array}{llrcrcrcrl} \min & f(x) = & 2x_1^2 + 2x_2^2 + 4x_1x_2 + x_1x_3 + x_2x_3 \\ \text{s.t.} & h_1(x) = & x_1^2 & + & x_2^2 & & & -1 & = 0 \\ & h_2(x) = & 2x_1 & + & 2x_2 & + & x_3 & -1 & = 0. \end{array}$$

This NLP is not too hard to solve by inspection: by substituting $x_3 = 1 - 2(x_1 + x_2)$, graphing the resulting 2-variable feasible region and lines of constant objective function, and observing the symmetry of the problem, it can be seen that there is a unique optimal solution given by $x = (-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 1 + 2\sqrt{2})^T$.

However, it is usually more difficult to solve such problems. We seek a solution by examining the Lagrangian function:

$$\begin{aligned} L(x,\eta) &= 2x_1^2+2x_2^2+4x_1x_2+x_1x_3+x_2x_3 \\ &+ \eta_1(x_1^2+x_2^2-1)+\eta_2(2x_1+2x_2+x_3-1). \end{aligned}$$

By Theorem 4, and assuming one of the constraint qualifications holds, (we will check this later), if x is a local optimum, it must be that $h_1(x) = h_2(x) = 0$ and there must exist η such that

$$\nabla_x L(x, \eta) = \begin{bmatrix} 4x_1 + 4x_2 + x_3 + 2\eta_1 x_1 + 2\eta_2 \\ 4x_2 + 4x_1 + x_3 + 2\eta_1 x_2 + 2\eta_2 \\ x_1 + x_2 + \eta_2 \end{bmatrix} = 0.$$

From the first two rows, we see that

$$4x_1 + 4x_2 + x_3 + 2\eta_1 x_1 + 2\eta_2 = 0 = 4x_2 + 4x_1 + x_3 + 2\eta_1 x_2 + 2\eta_2.$$

This tells us that $x_1 = x_2$.

By substituting $x_1 = x_2$ into $h_1(x) = x_1^2 + x_2^2 - 1 = 0$ we get $x_2 = \pm\frac{1}{\sqrt{2}}$, and hence $x_1 = \pm\frac{1}{\sqrt{2}}$.

Substituting these values into $h_2(x) = 2x_1 + 2x_2 + x_3 - 1 = 0$ yields $x_3 = 1 \mp 2\sqrt{2}$.

To find the optimal Lagrange multipliers corresponding to the two points found, we use the partial derivative of the Lagrangian with respect to x_3 to give $\eta_2 = -(x_1 + x_2) = \mp\sqrt{2}$ and the partial derivative with respect to x_1 to give $\eta_1 = -\frac{1}{2x_1}(4x_1 + 4x_2 + x_3) - \frac{\eta_2}{x_1} = \mp\frac{1}{\sqrt{2}}$.

To summarize, the NLP has two stationary points:

$$x = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 1 - 2\sqrt{2} \end{bmatrix} \quad \text{and} \quad x = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \\ 1 + 2\sqrt{2} \end{bmatrix}.$$

The Lagrange multipliers for $x = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 1 - 2\sqrt{2})^T$ are $\eta = (-\frac{1}{\sqrt{2}}, -\sqrt{2})$, and the Lagrange multipliers for $x = (-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 1 + 2\sqrt{2})^T$ are $\eta = (\frac{1}{\sqrt{2}}, \sqrt{2})$.